

Amendments to the Specification:

Please replace paragraph 1 with the following amended paragraphs:

This application claims priority under 35 U.S.C. 119(e)(1) to (i) U.S. Provisional Patent Application No. 60/439,981, filed January 14, 2003, which is related to United States Patent Application Serial No. 10/040,525, entitled INFORMATION RETRIEVAL SYSTEM INCLUDING VOICE BROWSER AND DATA CONVERSION SERVER, (ii) ~~to~~ United States Provisional Application No. 60/348,579, entitled DATA CONVERSION SERVER FOR VOICE BROWSING SYSTEM, and (iii) ~~to~~ United States Provisional Application No. 60/350,923, entitled MULTI-MODE PLATFORM FOR INFORMATION RETRIEVAL SYSTEM.

COMPUTER PROGRAM LISTING APPENDIX

A CD-ROM containing computer program listings identified as Appendices A - D has been filed in connection herewith (having a file entitled 10469052.doc thereon), the contents of which are hereby fully incorporated by reference.

Please replace paragraph 61 with the following amended paragraph:

FIG. 6 is a flow chart showing operation of the conversion server 150. A source code listing of a top-level convert routine forming part of an exemplary software implementation of the conversion operation illustrated by FIG. 6 is contained in Appendix A in the Computer Program Listing Appendix. In addition, Appendix B in the Computer Program Listing Appendix provides an example of conversion of a WML-based document into VoiceXML-based grammatical structure in accordance with the present invention.

Please replace paragraph 63 with the following amended paragraph:

Once the identified Web-based or other content has been retrieved by the retrieval module 310, the parser 340 is invoked to parse the retrieved content using the DTD applicable to the format of the retrieved content (step 616). In the event of a parsing error (step 618), an error message is returned (step 620) and processing is terminated (step 622). A root node of the DOM representation of the retrieved content generated by the parser 340, i.e., the parse tree, is then identified (step 623). The root node is then classified into one of a number of predefined classifications (step 624). In the exemplary embodiment each node of the parse tree is assigned to one of the following classifications: *Attribute, CDATA, Document Fragment, Document Type, Comment, Element, Entity Reference, Notation, Processing Instruction, Text*. The content of the root node is then processed in accordance with its assigned classification in the manner described below (step 628). If all nodes within two tree levels of the root node have not been processed (step 630), then the next node of the parse tree generated by the parser 340 is identified (step 634). If not, conversion of the desired portion of the retrieved content is deemed completed (step 631) and an output file containing such desired converted content is generated and the process is stopped (step 631).

Please replace paragraph 65 with the following amended paragraph:

Appendix C in the Computer Program Listing Appendix contains a source code listing for a TraverseNode function which implements various aspects of the node traversal and conversion functionality described with reference to FIG. 6. In addition, Appendix D in the Computer Program Listing Appendix includes a source code listing of a ConvertAtr function, and of a ConverTag function referenced by the TraverseNode function, which collectively operate to WML tags and attributes to corresponding VoiceXML tags and attributes.

Please replace paragraph 66 with the following amended paragraph:

FIGS. 7A and 7B are collectively a flowchart illustrating an exemplary process for transcoding a parse tree representation of an WML-based document into an output document comporting with the VoiceXML protocol. Although FIG. 7 describes the transcoding process with specific reference to the WML and VoiceXML protocols, the process is also readily applicable to conversion between other visual-based and voice-based protocols. In 702, a root node of the parse tree for the target WML document to be transcoded is retrieved. The type of the root node is then determined and, based upon this identified type, the root node is processed accordingly (step 731). Specifically, the conversion process determines whether the root node is an attribute node (step 706), a CDATA node (step 708), a document fragment node (step 710), a document type node (step 712), a comment node (step 714), an element node (step 716), an entity reference node (step 718), a notation node (step 720), a

processing instruction node (step 722), or a text node (step 724).

Please replace paragraph 72 with the following amended paragraph:

Referring again to FIG. 7B, any "child" tags of the Select tag are then processed as described above with respect to the original "root" node of the parse tree and accordingly converted into VoiceXML-based grammatical structures (step 740). Upon completion of the processing of each child of the Select tag, the information associated with the next unprocessed node of the parse tree is retrieved (step 744). To the extent an unprocessed node was identified in step 744 (step 746), the identified node is processed in the manner described above beginning with step 706. Otherwise, a converted voice XML is output (step 741) and the process stopped (step 743).

Please replace paragraph 128 with the following amended paragraphs:

In order to switch to voice mode when subsection-based switching is available as a selectable option, the following is required by MMGC 812

- 1) The VoiceXML compliant server.
- 2) The URL to be browsed in voice once the mode is switched to voice from visual.
- 3) Whether subsection-based switching is to be applied or not.

The VoiceXML gateway 902 is provided either by the carrier or the enterprise using the MMGC 812. The URL is furnished by the application developer. The URL could be in any of the visual based XML languages such as WML/CHTML/xHTML etc. or in voice based XML languages such as VoiceXML. The MMGC 812

identifies the type of the URL and then applies appropriate conversion rules to make the content suitable to be browsed using voice. The converter is used to convert the visual source into MultiModal VoiceXML, which provides the ability to come back to visuals depending on the user's choice.

Please replace paragraph 154 with the following amended paragraph:

FIG. 12 illustratively represents the manner in which the above source content would be rendered by a browser operative within the subscriber unit 802 (see boxes 1202, 1204, 1206). When the user selects LISTEN, visual source is accessed in voice through MMGC 812. The source is then converted by the MMGC 812 into MultiMode VoiceXML form, which includes a number of tags/marks:

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<catch event="show">
<prompt>You will be switched to visual mode in a moment</prompt>
<goto
next="http://MMGC_IPADDRESS/scripts/sendpush.Script?title=Alert&phoneNo=xxxxxxx
xxx&url=http://urladdressofthevisualpage&browsingpointer=expr(browsingpointer)"/>
</catch>
<var name="browsingpointer" value="0"/>
<form id="test">
    <block>
        <prompt>
            Following applications will demonstrate the power of
            V-Enable's MMGC.
        </prompt>
        <script>
            browsingpointer++;
        </script>

        <prompt>
            Select any of the following application and
            experience the difference.
```

```

        </prompt>
        <script>
            browsingpointer++;
        </script>
        <prompt>
            Multimode applications
        </prompt>
        <script>
            browsingpointer++;
        </script>
    </block>

    <field name="noname">
        <prompt>cnet news</prompt>
        <prompt> Please Say Okay or next.</prompt>
        <grammar mode="voice" xml:lang="en-US" version="1.0"
root="command1">
            <rule id="command1" scope="public">
                <ruleref uri="#action1"/>
            </rule>
            <rule id="action1" >
                <one-of>
                    <item>Okay</item>
                    <item>next</item>
                </one-of>
            </rule>
        </grammar>
        <filled>
            <if cond="NONAME0 == 'okay'">
                <goto
                    next="http://
http://MMGC_IPADDRESS/scripts/wml.Script?push=1&a
mp;phoneNo=xxxxxxxxxx&url=http://wap.cnet.com
"/>
                <else/>
                    <prompt> next </prompt>
                </if>
            </filled>
        </field>
        <script>
            browsingpointer++;
        </script>
        <field name="noname">
            <prompt>BBC news</prompt>
            <prompt> Please Say Okay or next.</prompt>
            <grammar mode="voice" xml:lang="en-US" version="1.0"
root="command1">
                <rule id="command1" scope="public">
                    <ruleref uri="#action1"/>
                </rule>
                <rule id="action1" >

```

```

        <one-of>
            <item>Okay</item>
            <item>next</item>
        </one-of>
    </rule>
</grammar>
<filled>
    <if cond="NONAME0 == 'okay'">
        <goto
            next="http://
http://MMGC_IPADDRESS/scripts/wml.Script?push=1&a
mp;phoneNo=xxxxxxxxxx&url=http://www.bbc.co.u
k/mobile"/>
        <else/>
            <prompt> next </prompt>
        </if>
    </filled>
</field>
<script>
    browsingpointer++;
</script>
<field name="noname">
    <prompt>Yahoo Email</prompt>
    <prompt> Please Say Okay or next.</prompt>
    <grammar mode="voice" xml:lang="en-US" version="1.0"
root="command1">
        <rule id="command1" scope="public">
            <ruleref uri="#action1"/>
        </rule>
        <rule id="action1" >
            <one-of>
                <item>Okay</item>
                <item>next</item>
            </one-of>
        </rule>
    </grammar>
    <filled>
        <if cond="NONAME0 == 'okay'">
            <goto
                next="http://
http://MMGC_IPADDRESS/scripts/wml.Script?push=1&a
mp;phoneNo=xxxxxxxxxx&url=http://wap.yahoo.co
m"/>
            <else/>
                <prompt> next </prompt>
            </if>
        </filled>
    </field>
    <script>
        browsingpointer++;
    </script>

```

```

<field name="noname">
  <prompt>Multimode Email</prompt>
  <prompt> Please Say Okay or next.</prompt>
  <grammar mode="voice" xml:lang="en-US" version="1.0"
root="command1">
    <rule id="command1" scope="public">
        <ruleref uri="#action1"/>
    </rule>
    <rule id="action1" >
        <one-of>
            <item>Okay</item>
        </one-of>
    </rule>
  </grammar>
  <filled>
    <if cond="NONAME0 == 'okay'">
      <goto
        next="http://
        http://MMGC_IPADDRESS/scripts/wml.Script?push=1&a
        mp;phoneNo=xxxxxxxxxx&url=http://MMGC_IPADDRE
        SS/appl/email.Script"/>
    </filled>
  </field>
</form>
</vxml>

```

Please replace paragraph 157 with the following amended paragraph:

FIG. 13 depicts the visual source seen by the user in response to the above source (see boxes 1302, 1304). Note that the "ViewAll" softkey allows user to see the original content.

Please replace paragraph 159 with the following amended paragraph:

FIG. 14 shows the corresponding visual source seen by the user via the screen of the applicable subscriber unit 802 (see box 1402). Note that the "ViewAll" softkey allows user to see the original content.

Please delete pages 99-112 of the specification.